**NodeJS Capstone Project**

**Project:** Build an API using NodeJS

**Industry:** General

**Problem Statement:**

Your company wants you to build an API which can sign up and sign in users based on data

provided by the users when using the API

**Description:** In this project we will use ExpressJS to build and serve the API. The API needs to be able to accept user data and store it in a mongodb server. At the time of sign up check if the user is unique and at the time of sign in check if username and password match and are available in the database.

**Tasks to be performed**:

Initialize an empty npm project

Install express and a mongodb package (either mongodb or mongoose)

Create routes for sign up and sign in with POST as request method

In sign up check if user does not already exist in the database. If not then check if

password is at least 5 characters long and add it to the database.

If either or both condition returns false, then return an error to the user with the

appropriate error messages.

Make sure to hash the password before storing them in the database

(use the bcrypt package)

During sign in check if username exists and then check if the given password

matches. If it does return a randomly generated token of 64length

If username does not exist or password is incorrect return an error with 404 status

code and message incorrect credentials

NOTE: When comparing passwords make sure you are comparing the hashed version

of passwords with each other.

Solutions

**Step1: npm init --y or npm init**

**Step2: npm install express body-parser ejs mongoose**

**Step3: setting up folder structure for project**

create a folder views and make below ejs file

**home.ejs**

**login.ejs**

**signup.ejs**

**companyPage.ejs**

**submit.ejs**

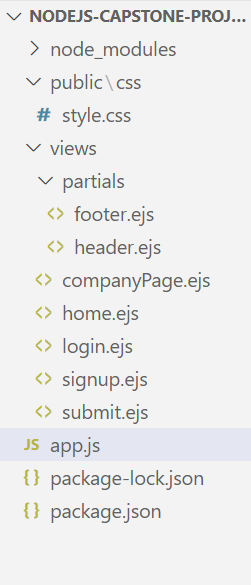
Also inside views folder make another folder partials and create below ejs file

**header.ejs**

**footer.ejs**

Now for styling above all ejs file , create a folder public , inside public create another folder css and make **style.css** file

Now to create server file of our project make **app.js** file. The folder structure will look as shown below.



**Step4.1: Add below code in** app.js file

app.js

const express = require("express");

const bodyParser = require("body-parser");

const ejs = require("ejs");

var app = express();

app.set("view engine","ejs");

app.use(express.urlencoded({extended:true}));

app.use(express.static('public'));

app.get("/",function(req,res){

    res.render("home");

});

app.get("/login",function(req,res){

    res.render("login");

});

app.get("/signup",function(req,res){

    res.render("signup");

});

app.listen(5000,function(){

    console.log("Server started at port 5000");

});

**Step4.2: Add below code in** header.ejs file

<!DOCTYPE html>

<html lang="en" dir="ltr">

    <head>

        <meta charset="utf-8">

        <title>Secrets</title>

        <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.6.3/css/all.css"

            integrity="sha384-UHRtZLI+pbxtHCWp1t77Bi1L4ZtiqrqD80Kn4Z8NTSRyMA2Fd33n5dQ8lWUE00s/" crossorigin="anonymous">

        <link href="http://stackpath.bootstrapcdn.com/bootstrap/4.2.1/css/bootstrap.min.css" rel="stylesheet">

        <link rel="stylesheet" href="css/style.css">

    </head>

    <body>

**Step4.3: Add below code in** footer.ejs file

</body>

</html>

**Step4.4: Add below code in** home.ejs file

<%- include('partials/header') %>

<div class="jumbotron centered">

    <div class="container">

        <i class="fa fa-user fa-6x"></i>

        <h1 class="display-3">Home Page</h1>

        <p class="lead">Existing Users can login to their account. New Users can create account</p>

        <h4>Please click on Login button. Don't have a account! click on Sign Up button</h4>

        <hr>

        <a class="btn btn-dark btn-lg" href="/login" role="button">Login</a>

        <a class="btn btn-light btn-lg" href="/signup" role="button">Sign Up</a>

    </div>

</div>

<%- include('partials/footer') %>

**Step4.5: Add below code in** signup.ejs file

<%- include('partials/header') %>

<div class="container mt-5">

    <h1>Register</h1>

    <div class="row">

        <div class="col-sm-8">

            <div class="card">

                <div class="card-body">

                    <!-- Makes POST request to /signup route -->

                    <form action="/signup" method="POST">

                        <div class="form-group">

                            <label for="email">Email</label>

                            <input type="email" class="form-control" name="username">

                        </div>

                        <div class="form-group">

                            <label for="password">Password</label>

                            <input type="password" class="form-control" name="password">

                        </div>

                        <button type="submit" class="btn btn-dark">Sign Up</button>

                    </form>

                </div>

            </div>

        </div>

    </div>

</div>

**Step4.6: Add below code in** login.ejs file

<%- include('partials/header') %>

<div class="container mt-5">

    <h1>Login</h1>

    <div class="row">

        <div class="col-sm-8">

            <div class="card">

                <div class="card-body">

                    <!-- Makes POST request to /login route -->

                    <form action="/login" method="POST">

                        <div class="form-group">

                            <label for="email">Email</label>

                            <input type="email" class="form-control" name="username">

                        </div>

                        <div class="form-group">

                            <label for="password">Password</label>

                            <input type="password" class="form-control" name="password">

                        </div>

                        <button type="submit" class="btn btn-dark">Login</button>

                    </form>

                </div>

            </div>

        </div>

    </div>

</div>

**Step4.7: Add below code in** companyPage.ejs file

<%- include('partials/header') %>

<div class="jumbotron text-center">

    <div class="container">

        <i class="fas fa-key fa-6x"></i>

        <h1 class="display-3">You've landed to the Company Page</h1>

        <p class="secret-text">This page is Authenticated</p>

        <hr>

        <a class="btn btn-light btn-lg" href="/logout" role="button">Log Out</a>

        <a class="btn btn-dark btn-lg" href="/submit" role="button">Submit a Secret</a>

    </div>

</div>

<%- include('partials/footer') %>

**Step4.8: Add below code in** submit.ejs file

<%- include('partials/header') %>

<div class="container">

    <div class="jumbotron centered">

        <i class="fas fa-key fa-6x"></i>

        <h1 class="display-3">Company</h1>

        <p class="secret-text">Hi Welcome to our Company, share with your friends and family</p>

        <form action="/submit" method="POST">

            <div class="form-group">

                <input type="text" class="form-control text-center" name="secret" placeholder="What's your secret?">

            </div>

            <button type="submit" class="btn btn-dark">Submit</button>

        </form>

    </div>

</div>

<%- include('partials/footer') %>

**Step4.9: Add below code in** style.css file

body{

    background-color: #EBECEF;

}

.centered{

    padding-top: 200px;

    text-align: center;

}

.secret-text{

    text-align: center;

    font-size: 2rem;

    color: #fff;

    background-color: #000;

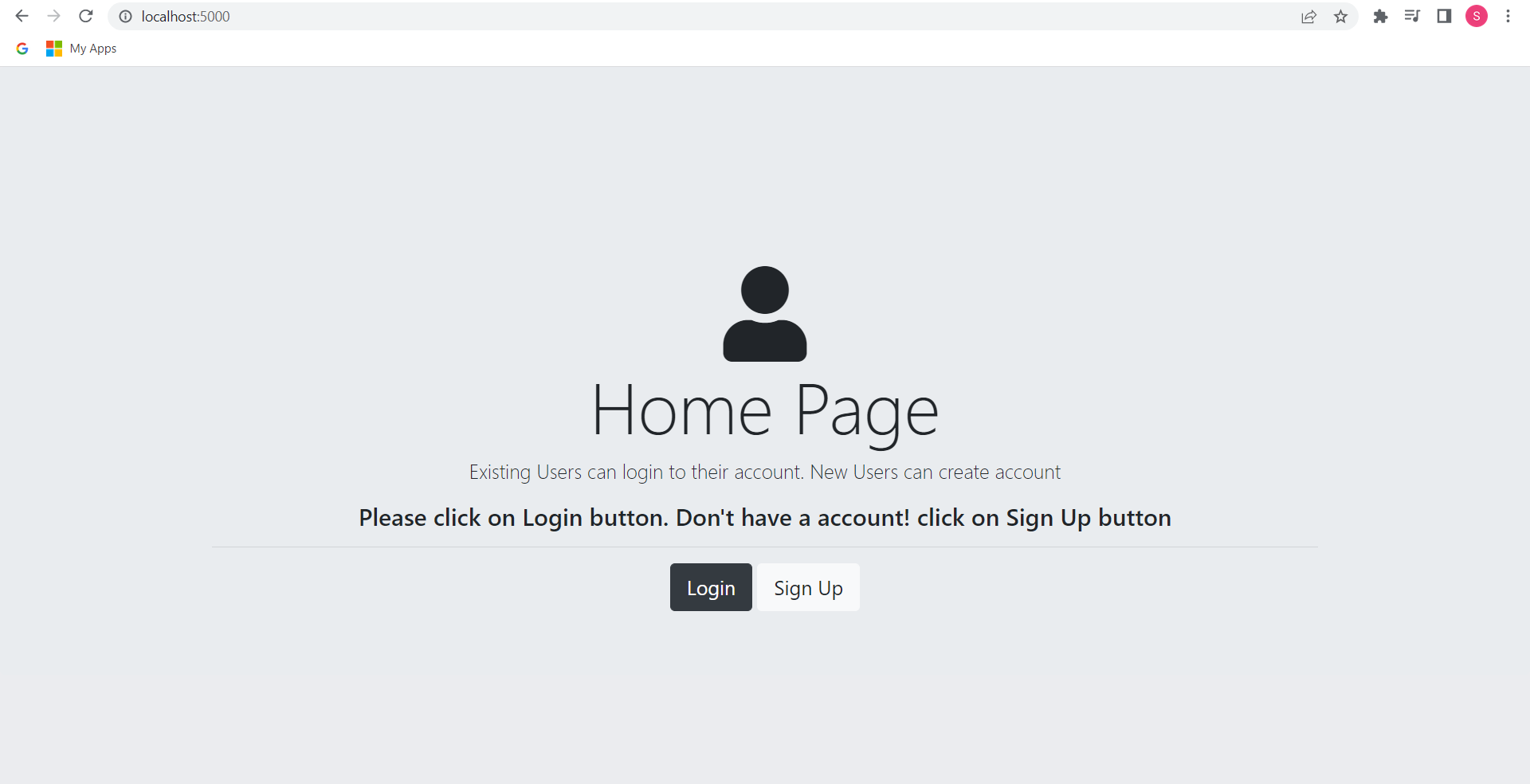
}

**Step5: Open the terminal and type below command**

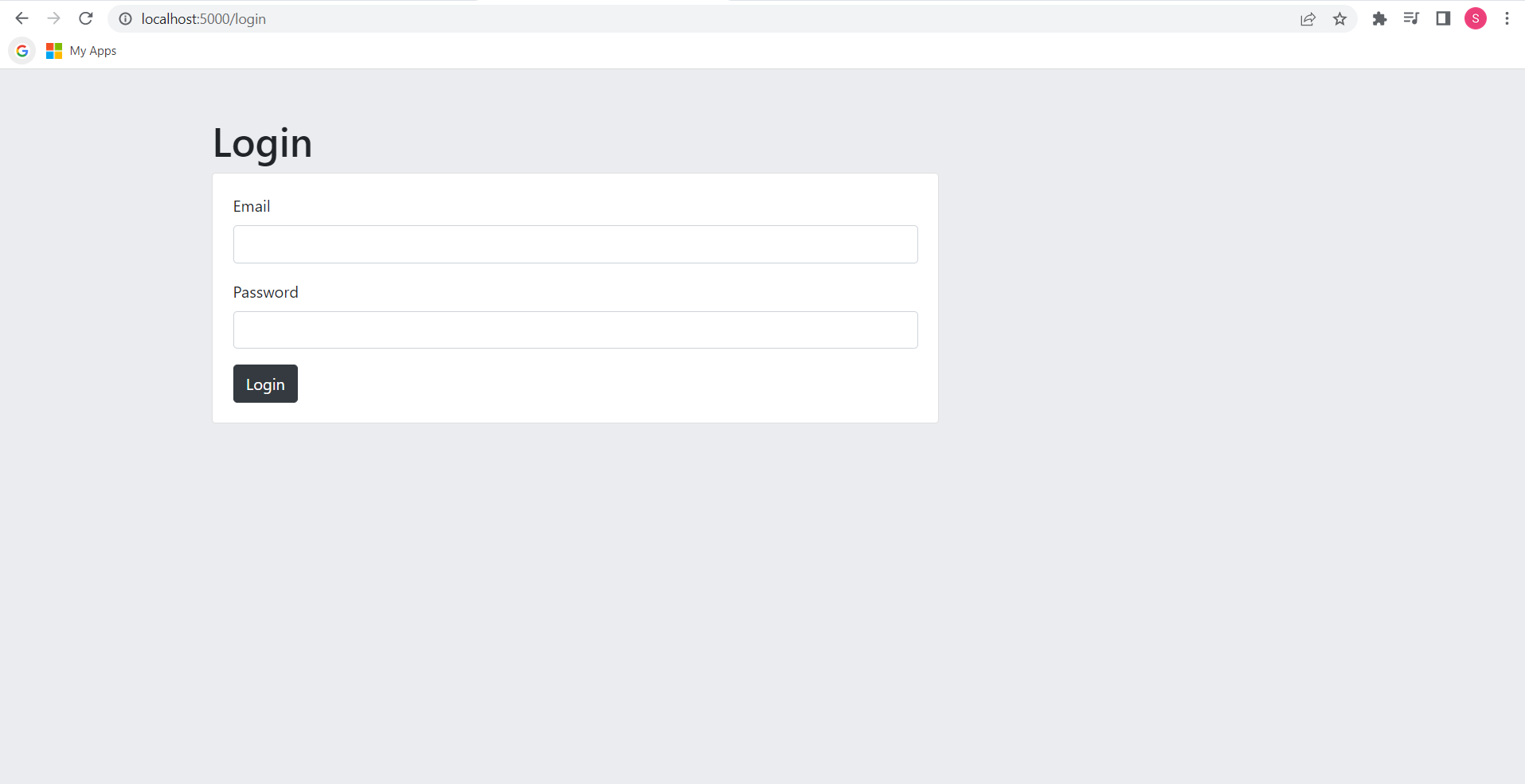
$ node app.js

Server started at port 5000

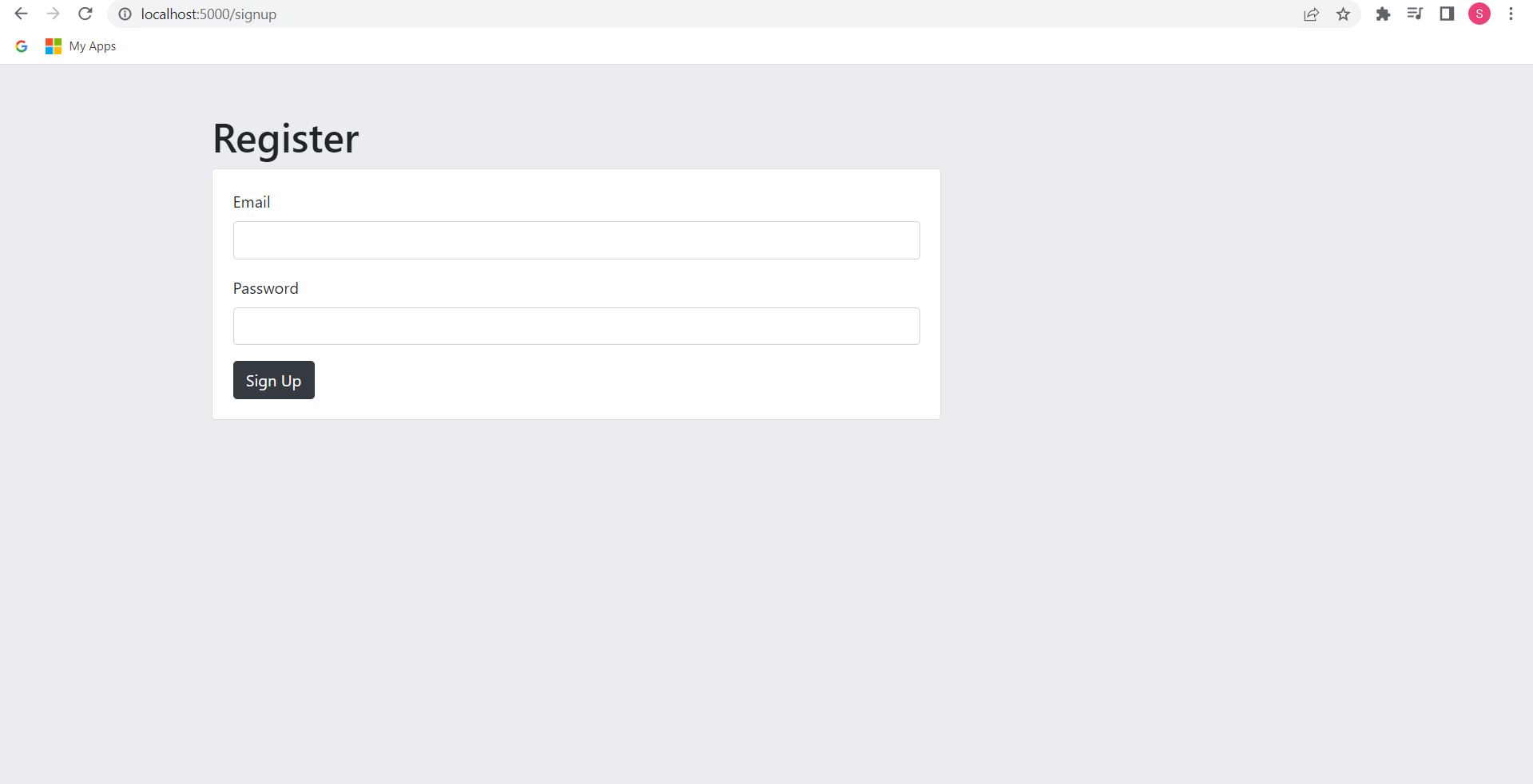
**Step6: Open the browser and type** [**http://localhost:5000/**](http://localhost:5000/) **then the home page will look like as show below**



**Step7:** When **we click on Login button then the page will look like as shown below**



**Step8: And when we click on Sign Up button then the page will look like as shown below**



**Step9: To** **enable password encryption we will use a npm package named as mongoose-encrytion**

Open the terminal and run below command

$ npm i mongoose-encryption

**Step10: Now the app.js file will be as coded below. Here we have ceated routes and function for login and sign up**

app.js

const express = require("express");

const bodyParser = require("body-parser");

const encrypt = require('mongoose-encryption');

const ejs = require("ejs");

var app = express();

app.set("view engine","ejs");

app.use(express.urlencoded({extended:true}));

app.use(express.static('public'));

//connect app.js with mongoDB server

const mongoose = require("mongoose");

mongoose.connect("mongodb://localhost:27017/users");

const trySchema = new mongoose.Schema({

    email: String,

    password: String

});

//for both encryption and decryption , we need to create a key

const secret = "thisismykey.";

trySchema.plugin(encrypt,{secret:secret,encryptedFields:["password"]});

const item = mongoose.model("second",trySchema);

app.get("/",function(req,res){

    res.render("home");

});

app.post("/signup",function(req,res){

    const newUser = new item({

        email: req.body.username,

        password: req.body.password

    });

    newUser.save(function(err){

        if(err){

            console.log(err);

        }

        else{

            res.render("companyPage")

        }

    });

});

app.post("/login",function(req,res){

    const username = req.body.username;

    const password = req.body.password;

    item.findOne({email:username},function(err,foundUser){

        if(err){

            console.log(err);

        }

        else{

            if(foundUser){

                if(foundUser.password === password){

                    res.render("companyPage");

                }

            }

        }

    });

});

/\*app.get("/",function(req,res){

    res.render("home");

});\*/

app.get("/login",function(req,res){

    res.render("login");

});

app.get("/signup",function(req,res){

    res.render("signup");

});

app.listen(5000,function(){

    console.log("Server started at port 5000");

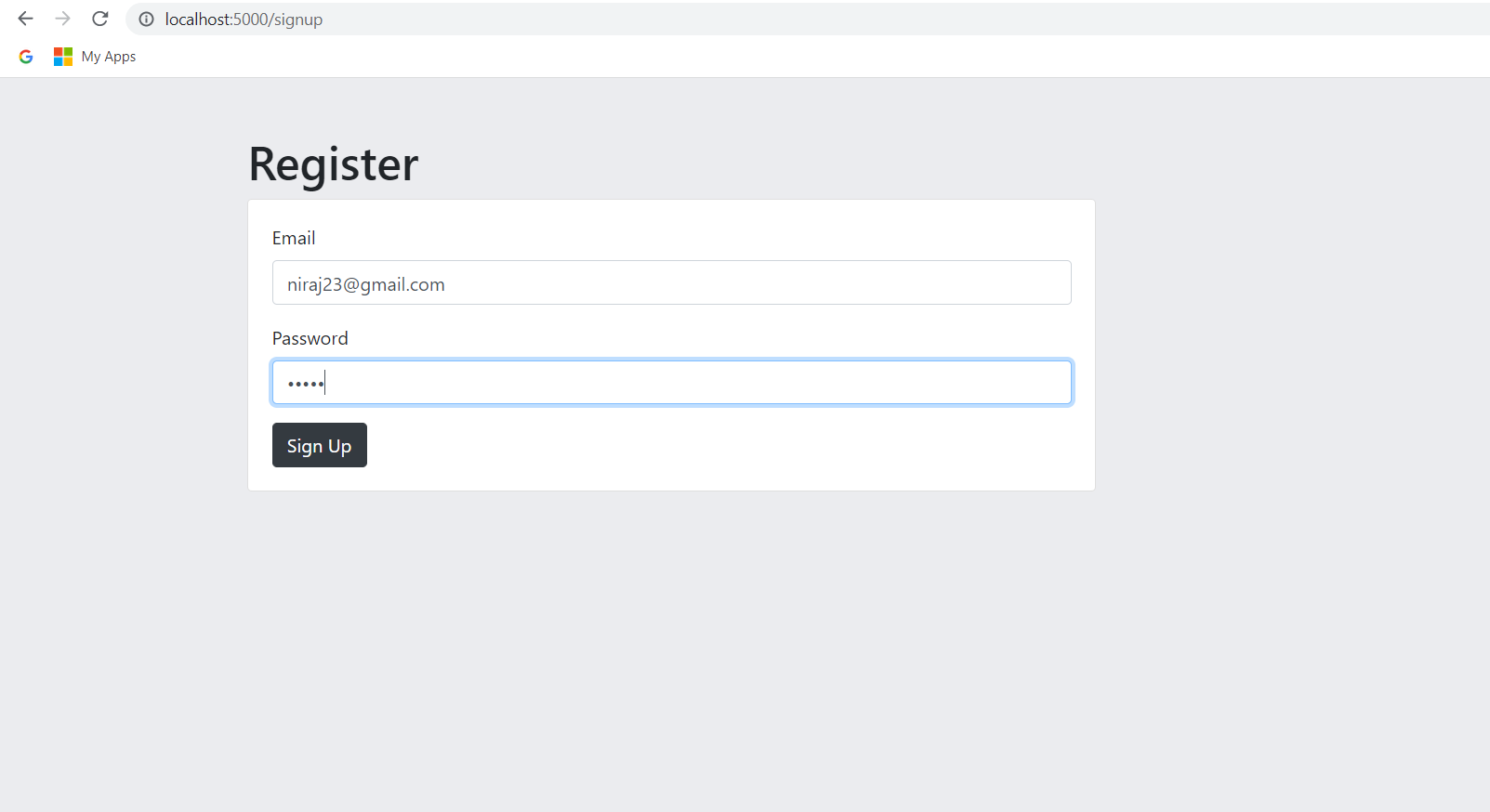
});

**Step11: Open the terminal and type below command**

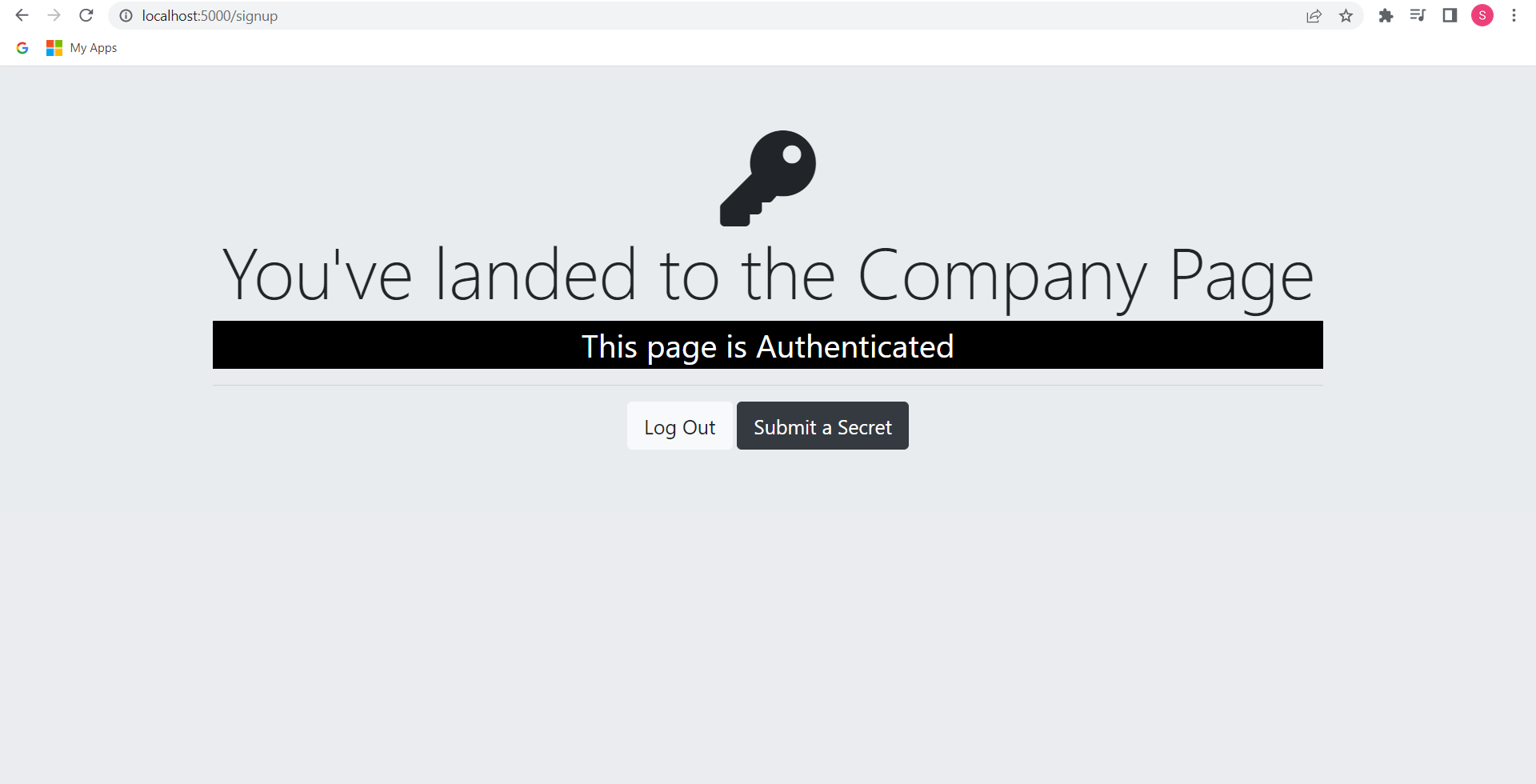
$ node app.js

Server started at port 5000

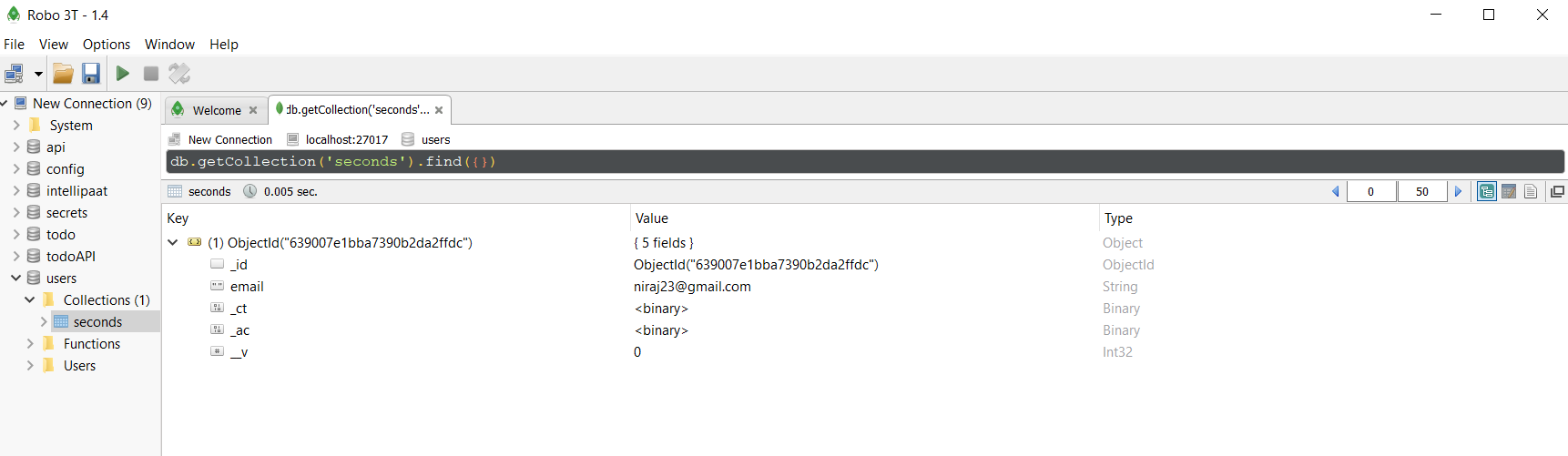
**Step12: Open the browser and type localhost:5000 and then click on Sign Up button and fill the details as shown below**

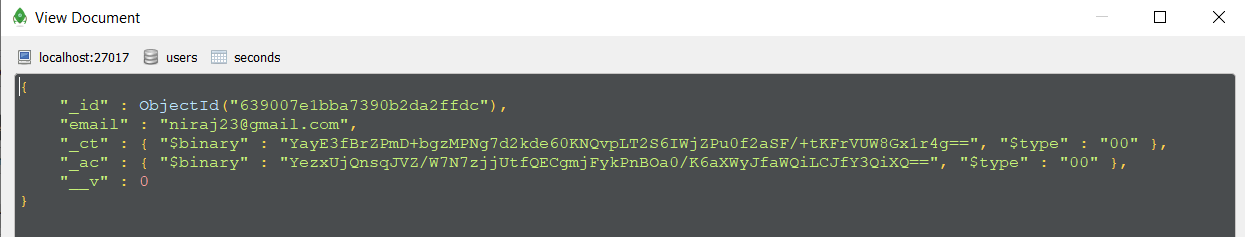


**Step13: After filling details like Email and Password , click on Sign Up button then it will redirect to another page named as companyPage as sow below**

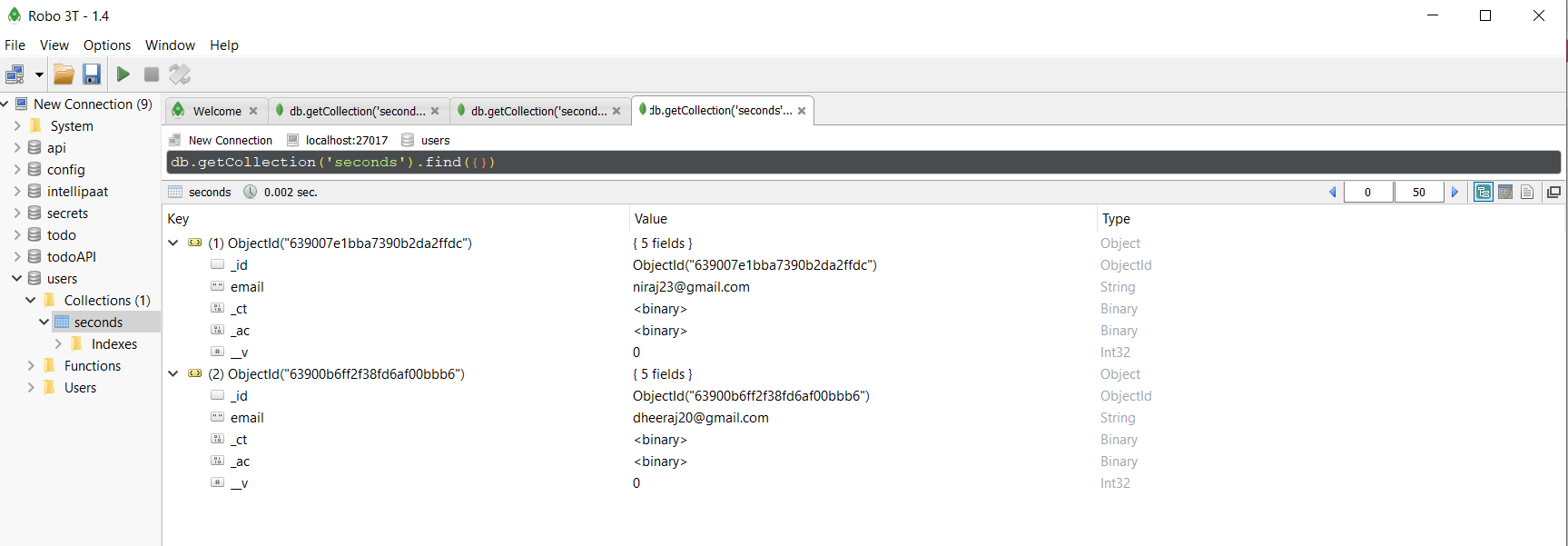


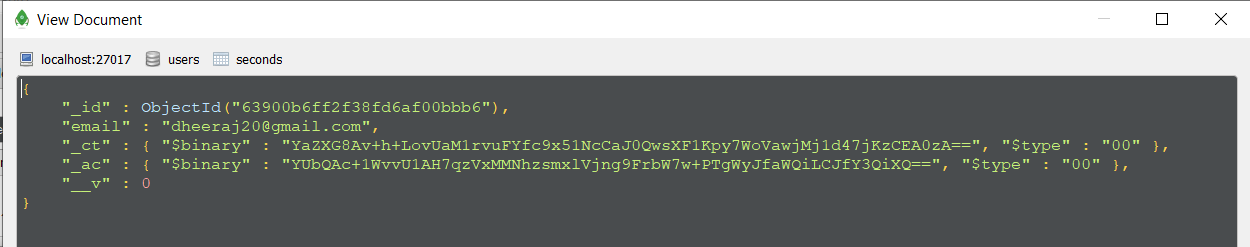
**Step14: Now we will check the database whether this user details saved or not. And we can see that user details are present as shown below**





**Step15: We can also add more users. I have added one more user as shown below**





**Step16: Open the terminal and run below command.**

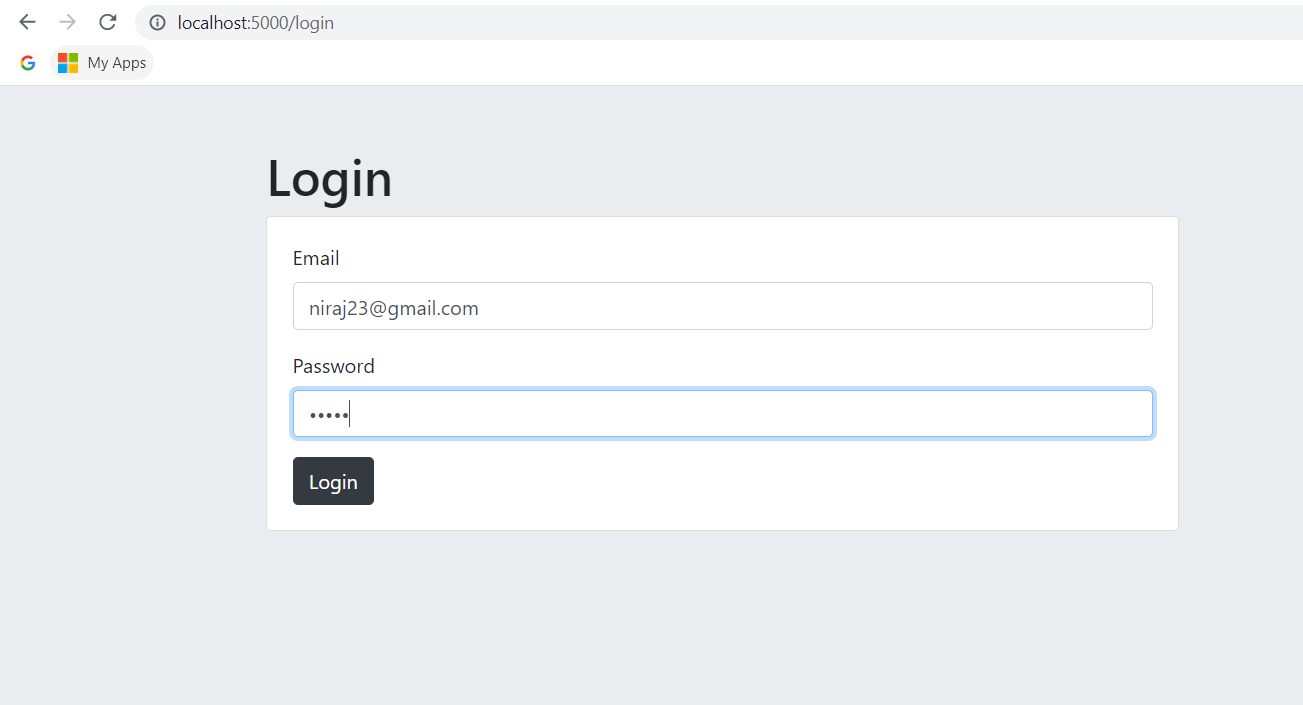
DELL@SujataKumari MINGW64 ~/Desktop/NodeJS/Project/NodeJS-Capstone-Project (master)

$ node app.js

Server started at port 5000

**Step17: Now we will check login credential**

**Open the browser and type localhost:5000 and then click on Login button. After clicking on Login button below page will open. Here we will login with the credential with which user is already registered.**



**Step18**: As soon as we click on login button show above , it will redirect us to a new page as shown below.

